A SZÁMITÁSTECHNIKA TANITÁSA A TECHNIKA ÓRÁKON

Taresay Tamás

A számitástechnika jelentősége az élet minden területén rohamosan növekszik. Ennek felismerése tette szükségessé, hogy e tudomány alapjait már a középiskolák tanulói is elsajátithassák. A korábbi években ennek az volt legnagyobb akadálya, hogy a középiskolák nem rendelkeztek számitógépekkel. Kormányunk határozata alapján a mult évben egy örvendetes folyamat kezdődött, amelynek eredményeként ma már minden középiskola rendelkezik legalább egy mikroszámitógéppel. Most uj probléma áll előttünk; meg kell találni a számitástechnika tanitásának helyét, felhasználásának lehetőségét a középiskolában. Kézenfekvő lenne a számitástechnikát a matematika órákon tanitani, de a matematika tananyaga nagyon bő, óraszáma - különösen az alaptantervben - kevés, ezért a jelenlegi keretek között erre nincs lehetőség.

E tanulmány célja bemutatni, hogy a gimnáziumokban a számitástechnika alapjait a technika tantárgy keretében is lehet tanitani, mert

a) a jelenlegi technika tanterv keretei megfelelnek e célnak. Ezt a továbbiakban részletesebben is ki fogjuk fejteni;
b) mivel a levelező oktatási formában kiképzett technika tanárok jó része matematika szakos is, a számitástechnika tanitásának személyi feltételei is általában biztositottak;
c) a számitástechnika különválasztása a matematikától e tudományok egymáshoz való viszonyát is helyesen tükrözi.

Javaslatunk lényege az, hogy a tananyag a számitástechnikára épüljön. Tanitsuk meg a technika fogalmait, törvényeit, és a feladatokat elsősorban a számitástechnika köréből vegyük.

Az első évben programozható zsebszámológép /pl. PTK 1050/ alkalmazását tartjuk szükségesnek. Ezen megtanitható a programozás technikája. A lépések korlátozott száma miatt a "takarékos" programirás is kifejleszthető. Még elfogadható ellátottságnak tartjuk, ha három tanulónak jut egy zsebszámológép. A második évben a BASIC nyelvet ismerő mikroszámitógép /ABC-80 HT-1080Z/ használata ajánlható. A grafikus üzemmód ezeket a gépeket alkalmassá teszi különböző modellezési feladatok, szemléltetési feladatok megvalósitására. A másodikos technika tananyagban, a fizikában is kiemelt szerepe és jelentősége van a modellezésnek. Szükségesnek tartjuk, hogy legalább három ilyen számitógép álljon rendelkezésre az iskolában. A technika tanterve, és a tankönyvben lévő tananyaga jelentős mértékben különbözik. A tantervben előirt időbeosztástól eltérő tervezést igényel a tankönyvi anyag elsajátittatása. Elképzeléseink, különösen az első osztályban, a tankönyvhöz jól kapcsolhatók.

Az 1982-83-as tanévben három első osztályban és egy matematika-angol tagozatos harmadik osztályban - kisérletképpen - kipróbáltuk az itt leirt eljárást. Ismertetjük a tanitott anyagot, beszámolunk a tanitás közben szerzett tapasztalatokról, és javaslatot teszünk a második év tananyagára is. Ezután függelékként [1] közlünk egy programgyüjteményt, amely felhasználható az oktatásban. E gyüjteményben játékprogramot nem szerepeltettünk, mert a tanulók hajlamosak arra, hogy a számitógépet kizárólag játéknak fogják fel.

Az elsős tananyag bevezetőjeként a *technika fogalmát* kell megtanitani. Hangsulyozzuk a technika funkcionális jellegét, mert ez a számitógép fogalmának tisztázásakor jó kapaszkodó lehet. A technika alapfogalmaként megismertetjük

a *technika rendszert*, annak általános feladatát. Ez a fogalom meglehetősen absztrakt, ezért célszerü példákkal megvilágitani. Egyik példaként a számitógépről beszélünk, tisztázzuk a számológép és a számitógép közötti alapvető különbséget. Ezek után rátérhetünk egy konkrét technikai rendszer vizsgálatára, ismertetjük a PTK 1050-es programozható zsebszámológépet, amit a tanulók a tanévben használni fognak. Először a mikroprocesszor fogalmáról szólunk [2], majd a felhasználó rendelkezésére álló memóriarekeszekről /regiszterekről/. Indokoljuk a gép elnevezését /50 programozási lépés/, a kalkulátorként való alkalmazását. Egyszerü számolási feladatok segitségével ismertetjük a kezelőgombok jelentését és gyakoroltatjuk alkalmazásukat, a regiszterek használatát, a müveletek prioritását. Szemléltetésként a tanulók kezébe adunk sokszorositott formában egy táblázatot, amely a kezelőgombok feliratát, angol elnevezését és annak jelentését tartalmazza. Ezek után rámutatunk arra, hogy a kalkulátor üzemmód csak az egyik felhasználási lehetőség, a gép ennél jóval többre képes, ugyanis programozható. Ennek tanitásakor hangsulyozni kell, hogy egy feladat számitógéppel való megoldhatóságának feltétele az, hogy matematikai formulákkal leirható legyen. Gépünk csak a kalkulátor üzemmódban megismert, apró lépések sorozatának elvégzésére képes. Az emberi gondolkodás viszont összetettebb, bonyolultabb ennél. A programirás lényege, hogy ezt a különbséget áthidalja, lebontsa a feladatot a gép lépéseinek egymásutánjára. Ezt általában két lépésben tesszük meg. Elősz̈r blokkvázlatot /folyamatábrát/ készitünk, majd annak alapján irjuk meg a *programot*. Megjegyezzük, hogy a technika tankönyv [7] a folyamatábra elnevezést más értelemben is használja. Mivel az órákon különböző képességü tanulókból álló csoportokkal foglalkozunk lényegesnek tartjuk, hogy minden feladat megoldását a gondolatmenetet rögzitő blokkvázlattal kezdjük. Ezek elemeit a másodikos tankönyv [7] végén a függelékben találhatjuk. Az első osztályban ezek

közül csak a következőket szükséges megtanitani:

a program eleje, a program véges, a müvelet, a döntés, a csomópont, az input /output/, az alprogram /szubrutin/.

Ezek után egyszerü folyamatábrák közös elkészitése következik. Első feladatnak a Fibonacci sorozat elemeinek kiiratását ajánljuk, ezt követi a pozitiv egész számok faktoriálisának számitása. A PTK 1050-es zsebszámológép programozási üzemmódjának ismertetésekor az utasitásokat a tanult blokkokkal együtt tárgyalhatjuk. E módszer megkönnyiti a blokkvázlat alapján történő programirás elsajátittatását. A továbblépés az utasitások ismerete nélkül lehetetlen, ezért a számonkérés ebben a szakaszban elengedhetetlen. Minden tanulótól meg kell követelni az utasitásoknak készség szinten való ismeretét. Csak ezután térhetünk rá az egyszerü programozási feladatok megoldására. Először a korábban elkészitett blokkvázlatok alapján irunk programot. Kezdettől fogva hangsulyozzuk, hogy az adott feladat megoldására többféle algoritmus létezhet, többféle blokkvázlat készithető, és egy adott blokkvázlat is több program forrása lehet. Az általános iskolát végzett tanulók számára ez nehezen fogadható el, mert hozzászoktak ahhoz, hogy egy feladatnak egyetlen helyes megoldása lehet. Elkészült programot csak akkor fogadunk el helyesnek, ha már a futtatás eredménye is a kivánalmaknak megfelelő. Fel kell hivni a tanulók figyelmét arra is, hogy helyes algoritmus alapján készült program a gép müködési tulajdonságai miatt rossz eredményt is adhat. Az elkészült programokat összehasonlitjuk, megállapithatjuk előnyeiket, hátrányaikat. Ez a vizsgálat dönt arról, hogy az adott feladat megoldásakor melyiket részesitjük előnyben. Ettől kezdve a komplex munkák keretében a tanulók programozási feladatokat kapnak. Kezdetben közös munkával, majd kisebb csoportokban készitik el a programot. A csoportok kialakitásánál rendező elv lehet az, hogy vagy az azonos tudásu tanulók dolgoznak együtt, vagy a különböző tudás-

szintü tanulók együttmöködését igényeljük. Az első esetben a jobb képességü és szorgalmu tanulók fejlődése biztositott, a gyengébbek ösztönzése elmarad. A második esetben az utóbbiak egy-két magasabb szinten álló tanuló irányitásával jól fejlődhetnek. A számonkéréseknél ügyeljünk arra, hogy az adott csoport minden tagja megértse az elkészült programokat. Továbblépés csak szilárd alapról történhet. Ha lemaradást tapasztalunk megállunk, a jobb képességüeknek külön feladatot adunk, és a gyengébbekkel is megértetjük a problémás részeket.

A függelékben hozzávetőlegesen növekvő nehézségi fok szerint, olyan programok találhatók, amelyek az órán felhasználhatók. Igy a tanulók tudásszintjének, érdeklődési körének megfelelően lehet választani. A függelékben megtalálható értékelő megjegyzések ugyancsak e választás megkönnyitését célozzák.

A számitástechnika további vizsgálata indokolja a *technikai rendszerek* fejlődésével kapcsolatos témakör előbbre hozását. A jelenlegi tanterv szerint ez a második osztályban tanitandó. Miután a tankönyvben [7] levő vázlat alapján megtárgyaltuk a technikai rendszerek iterativ fejlődésének sajátosságait, példaként a számitástechnika fejlődésének történetéről beszélünk [5]. Jól szemléltethetjük a tudományok s a technika fejlődésének kapcsolatát, megmutatjuk, hogy a társadalom és a technika fejlődése kölcsönhatásban van egymással.

Témánk szempontjából e vizsgálat fő jelentősége az, hogy eljuthatunk a *mai számitógépek szerkezetének, müködésének* megismertetéséhez. A számitógépek felépitése /hardware/ témánk szempontjából annyiban érdekes, amennyiben ezeket az ismereteket a programok irásakor felhasználhatjuk. Mindenképpen beszélni kell a számitógépek alapvető funkcionális egységeiről [9], azok fajtáiról, feladataikról. A Neumann-féle tárolt programozás elve csak ugy érthető meg, ha a központi memória felépitése vázlatosan ismert. Ebből

következik, hogy ennek tárgyalására a többi egységnél nagyobb sulyt kell fektetni.

Célszerünek tartjuk, ha a gimnáziumi oktatásban is használjuk a Kalmár Lászlótól származó fiktiv számitógép módszerét, vagy ennek egyszerüsitett változatát. Fiktiv gépünk memóriáját 1000 db memóriarekesz alkotja. Mindegyikben egy tizes számrendszerbeli, nyolcjegyü szám tárolható. Minden regiszternek sorszáma /cimkéje/ van /000-499/. A memória két részre osztható, 000-499-ig az utasitásokat, 500-999-ig az adatokat tárolja. A továbblépéshez ennyi ismeret is elegendő. Ha a tanulókat ez a kérdéskör mélyebben érdekli, akkor mód van a részletesebb tárgyalásra is. Az irodalomjegyzékben szereplő források ehhez is segitséget nyujthatnak [2], [5], [9].

A tanterv következő része a *technikai rendszer* és a *környezet kapcsolatával* foglalkozik. Kézenfekvő tehát, hogy ehhez igazodva a *számitógép és környezete* kapcsolatát vizsgáljuk. Ennek részeként először az ember és a gép közötti kommunikáció lehetőségeivel foglalkozzunk. A korábban tanult fiktiv memória segitségével a *tárolt programozás elvét* tárgyaljuk.

A programok utasitásai a számunkra fenntartott helyen, nyolcjegyü számok formájában találhatók. Egy utasitásnak megfelelő szám három részből áll. Az első két számjegy a müveleti kód, ezt követi két háromjegyü szakasz, amelyek a legtöbb esetben cimkét jelentenek. A müveleti kód megadja az elvégzendő müvelet fajtáját, és azt, hogy a müvelet eredményét melyik rekeszben kell tárolni. A pontos jelentését a géphez tartozó müveleti kódok táblázatából ismerhetjük meg. A fiktiv gépünkhöz tartozó táblázat egy részlete:

00 összeadás
01 összeadás
02 szorzás
03 szorzás
04 vezérlés átadás az 1. cimkére

05 ha az 1. cimkén levő szám egyenlő a második cimkén levővel.

06 Stop

.

.

Az ilyen táblázat segitségével elkészitett, az adott gépre irott, számokból álló utasitássort nevezzük *gépi kódu programnak*. Egyszerü példaként elkészithetjük az 5! kiszámitását elvégző gépi kódu programot. Egyetlen ilyen program tanulmányozása elegendő ahhoz, hogy a tanulók felismerjék a gépi kód hátrányait, nehézségeit. Ezután már magától értetődik a magasabb szintü programnyelvek s velük kapcsolatban a forditóprogram lényegének emlitése. A számitógép és a környezet kapcsolatát vizsgálva tovább, beszélünk a számitógépek megfelelő müködéséhez szükséges környezeti feltételekről, amelyeket - különösen a régebbi tipusu gépeknél - biztositani kell. E téma tárgyalását célszerü gépterem látogatással összekapcsolni, ahol a tanulók megismerhetik a klimaberendezés, az álpadló, az álmennyezet célját, rendeltetését.

Ezután a Mérés cimü fejezet következik. E témát véleményünk szerint két ponton kapcsolhatjuk a számitástechnikához:

a/ beszélhetünk a különböző bonyolult, tartós, sok változót vizsgáló mérések számitógépes feldolgozásáról, kiértékeléséről, és

b/ a tudományos célu mérések hibaszámitási programjának elkészitéséről.

Az erősitőkről szóló fejezet tanitásakor csupán a komplex munkák keretében foglalkozhatunk programozással.

A technikai rendszerek irányitása cimü rész általánosságban történő ismertetése után a *számitógépek irányitásával* foglalkozunk. A funkcionális egységek tárgyalása közben emlitést tettünk a vezérlő egységről, most részletesebben vizsgáljuk a programok futásakor betöltött szerepét. Jól

használható a tanitás során a Melléklet 8. blokkvázlata. Természetesen itt is csak a fiktiv gépre szoritkozunk.

Ezzel az első osztályos tananyag végére értünk. Fontosnak tartjuk, hogy mig tanév közben az általános technika és a számitástechnika anyagának feldolgozása párhuzamosan történt, az év végi rendszerező ismétlés keretében elkülönitve foglaljuk össze ezeket az ismereteket. Ebben az áttekintésben különös hangsullyal foglalkozzunk azokkal a számitástechnikai kérdéskörökkel, amelyeket a következő évben is felhasználunk. A nyári szünetre önálló feladattal is elláthatjuk a tanulókat. Az év közben kialakult érdeklődés a biztositéka annak, hogy néhányan érdemlegesen foglalkoznak is a problémákkal.

Az első osztály anyagának tanitásakor tapasztaltuk, hogy a számonkérés módszere lényeges a számitástechnika tanitásánál is. Fontos, hogy a tanulóktól a tényanyagon kivül csak az órákon elkészült vagy megbeszélt programok megértését kérjük. Az önálló programkészitést majd csak a fakultativ csoportokban állithatjuk követelményként a tanulók elé. Az óráinkon, tapasztalataink szerint, épithetünk tanitványaink öntevékenységére is. Szorgalmi feladatokat is adhatunk, amelyekkel a tanulók szivesen foglalkoznak. Önálló tevékenységükre épitve feladathatjuk az órán készült programok továbbfejlesztését, vagy röviditését, más algoritmusok keresését és önálló programirást is. A magasabb szintü programozási nyelvek emlitésekor néhány tanulóban kiváncsiság ébredhet e nyelveken történő programozás iránt. Ezt az érdeklődést meg kell ragadni, ilyen irányu igényeik kielégitésére célszerü szakkört szervezni. Ez természetesen előrevetiti annak szükségességét, hogy a második évben differenciált oktatást alkalmazzunk.

A másodikos tankönyv anyagának jó része ismétlő jellegü. Sok olyan ismeretet tartalmaz, amiket már az előző évben megtanitottunk. Ebből következik, hogy ha az első osztályban sikerült szilárd ismereteket nyujtanunk, akkor le-

hetőség nyilik a számitástechnika további tanitására is.

A másodikos tananyaghoz már nehéz olyan szervesen illeszteni a számitástechnikát, mint ahogy ez az első osztályban - véleményünk szerint - sikerült. Igy csak bizonyos kapcsolódási pontokat lehet megemliteni. A komplex munkák keretében a BASIC nyelven irathatunk programokat, miután a nyelv fontosabb utasitásait az elméleti órákon megtanitottuk. Ügyelni kell arra, hogy az ABC-80 és a HT gépek BASIC interpreterje néhány helyen eltér egymástól! Ha mindkét számitógéptipus megtalálható az adott iskolában, akkor célszerü együtt tanitani a kétfajta utasitáskészletét. A komplex munkák elején a gépek kezelését tanitjuk meg, és csak ezután következhet a programirás. Ajánlatos olyan feladatokat adni, amilyeneket már PTK 1050-nel megoldottunk, mert igy a tanulók leszürhetik a zsebszámológép és a mikroszámitógép közötti lényeges különbségeket. Lehetőséget kell biztositani arra is, hogy a tanulók tapasztalhassák azt, hogy a személyi számitógépek jóval többre képesek, mint a programozható zsebszámológépek. A feladatok kiválasztásakor segitséget nyujthatnak a függelékben közölt BASIC programok is.

A technika tankönyv [7] anyaga és a számitástechnika kapcsolódási pontjai közül néhányat emlitünk a következőkben.

A szabványositással kapcsolatban áttekinthetjük a blokkvázlatok szabványos elemeit, amelyek a tankönyv függelékében találhatók. Első osztályban ezek közül csak néhányat tárgyaltunk. Másodévre a tanulók hardware ismereteinek gyarapodása alapul szolgál a további blokkok megismertetéséhez.

Az ember és a gép kapcsolata cimü témát tekintve érdemes szót ejteni az ember és számitógép kapcsolatáról szóló, szélsőséges nézetek birálatáról. Meg kell állapitani, hogy a számitógép az emberi gondolkodást segitő rendkivül hasznos eszköz, de sohasem fogja helyettesiteni, feleslegessé tenni az embert.

A tankönyv több fejezete foglalkozik tervezéssel. Ide lehet kapcsolni a különböző számitógépes tervezési eljárásoknak /hálótervezés, szállitás tervezés/ az ismertetését.

A *modell* nagyon lényeges fogalom a technika anyagában. A technika tankönyv [7] foglalkozik a számitógépes modellek jelentőségével is. Több modellprogramot készittethetünk a tanulókkal a PTK 1050-re is és BASIC nyelven is. Személyi számitógépeink grafikus üzemmódját itt jól kamatoztathatjuk.

Az információs rendszerekkel kapcsolatos témakör kapcsán sort kerithetünk a *számitógépes információtárolás*, feldolgozás megemlitésére. A könyv utolsó fejezete a számitógépről szól. Ezt a rendszerező ismétlésnél olvasmányként használhatjuk fel. Szem előtt tartva a tanulók eltérő érdeklődését, esetleges további szándékait a számitástechnikával kapcsolatban.

Tanitási tapasztalatainkra támaszkodva megállapithatjuk, hogy a technika órákon - a jelenlegi tanterv anyagát szem előtt tartva - van lehetőség a számitástechnika oktatására, és a megszerzett ismeretek jól szolgálhatják a tanulók felsőfoku tanulmányainak előkészitését is. Végezetül még egyszer hangsulyozzuk, hogy az itt leirtak csak az első két tanév számitástechnika anyagát tartalmazták. Szakkörben, fakultativ blokkban ennek jelentős továbbfejlesztésére van lehetőség.

FÜGGELÉK

Ebben a részben a tanórákon tárgyalható programokat közlünk, amelyek a PTK 1050-es zsebszámológépre, valamint az ABC-80-as számitógépre készültek. Ez utóbbi gépre iródott programok egyszerüen átirhatók a HT 2080-as vagy a HT 1080-as számitógépekre.
Megjegyezzük, hogy a PTK 1050 programjai könnyüszerrel átirhatók BASIC nyelvre, erre is mutatunk néhány példát.

A programok leirását legtöbb esetben a blokkvázlatok megadásával kezdjük. Ezután következik a program, majd néhány megjegyzés, amelyeknek célja a programok értékelése, továbbfejlesztési lehetőségek feltárása, módszertani javaslatok a tanitáshoz.

A PTK 1050-es programok elején megadjuk a regiszterek kiosztását a változók között; aláhuzással jelöljük azokat a változókat, melyeknek a program futtatása előtt kezdőértéket kell adni; ahol a kezdeti érték állandó, azt az aláhuzás alatt rögzitjük. Ezek után következik a program, majd az első inditás /a kezdeti érték megadásával/ és végül az ujrainditás megmutatása.

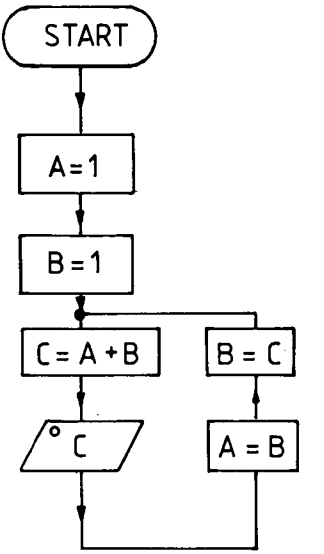
A Fibonacci sorozat elemeinek megadása
a rekurziv definició alapján [8]

a./

M∅	M1	M2
A	B	C
1	1	
	LRN	
∅∅	RCL	∅
	+	
	RCL	1
	=	
	STO	2
∅5	2nd	Pause
	RCL	1
	STO	∅
	RCL	2
	STO	1
1∅	RST	
	LRN	

b./

M∅	M1	M2
A	B	C
1	1	
	LRN	
∅∅	RCL	∅
	STO	2
	RCL	1
	SUM	2
	RCL	2
∅5	2nd	Pause
	RCL	1
	STO	∅
	RCL	2
	STO	1
1∅	RST	
	LRN	



(M. 1. 1.)

c./

M∅	M1
A	B
1	1
LRN	
∅∅	RCL ∅
	+
	RCL 1
	=
	2nd Pause
∅5	2nd Exc 1
	2nd Exc ∅
	RST
	LRN

Mindhárom esetben az inditás és az ujrainditás:

```
1
STO ∅
STO 1
RST
R/S
```

A feladat kapcsán megtárgyalhatjuk a végtelen ciklus fogalmát. Azt is láthatjuk, hogy a ciklus csak elvileg végtelen, hiszen ha elérjük a tárolható maximális számot, hibajelzéssel leáll a program futása. Az a/ és b/ változat teljesen egyenértékü, csak a felhasznált lépések különböznek. A c/ változat kevesebb memóriát igényel, és kevesebb programlépést használ. A kijelző regiszter és a memóriarekeszek közötti cserét lehet gyakoroltatni vele.

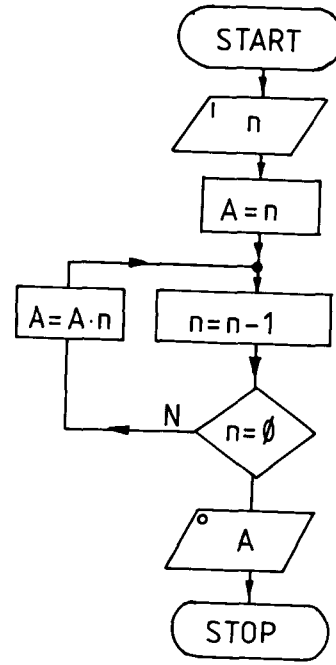
Továbbfejlesztési lehetőségek:

1. A gép jelezze ki a sorozat első két elemét is.
2. A sorozat kivánt számu elemét jelezze ki a gép.
3. A és B kezdőértékét a gép adja.

Az n! kiszámitása [4]

```
   MØ  M1
   M   A
     LRN
ØØ  2nd Dsz
    GTO 1
    RCL 1
    RIS
    2nd LbI 1
ØS  RCL Ø
    2nd Prd 1
    RST
    LRN
```

Inditás és ujrainditás: n
STO Ø
STO 1
RST
R/S



(M. 1. 2.)

A program csökkenő sorrendben szorozza össze a tényezőket. Csak két regisztert használunk, ami nagy jelentőségü olyan gépeknél, ahol kevés memóriarekesz áll rendelkezésre. Hátránya a programnak, hogy a Ø1 értékére helytelen eredményt ad. Segitségével gyakoroltathatjuk a Dsz utasitással történő ciklusszervezést.

Továbbfejlesztési lehetőség:

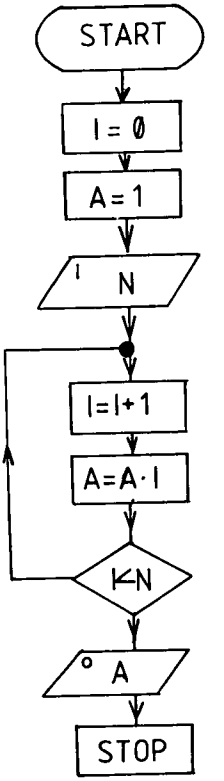
1. Ø! értékét is számittassuk ki.

2. Vizsgáltassuk meg, hogy n-nek természetes számot adtunk-e. Ellenkező esetben a gép adjon hibajelzést.

Az n! kiszámitása [4]

M0	M1	...	M7
I	A	...	N
0	1		

```
      LRN
00    1
      SUM 0
      RCL 0
      2nd Prd 1
      2nd INV x≥t
05    RST
      RCL 1
      R/S
      LRN
```



Inditás és ujrainditás: N
STO 7
0
STO 0
1
STO 1
RST
R/S

(M.1.3)

A program 1-gyel több regiszter használ, mint az előző változat, bonyolultabb az inditása is. Előnye, hogy a 0! értékét helyesen adja. A feltételes elágaztatás ujabb módját gyakoroltathatjuk vele.

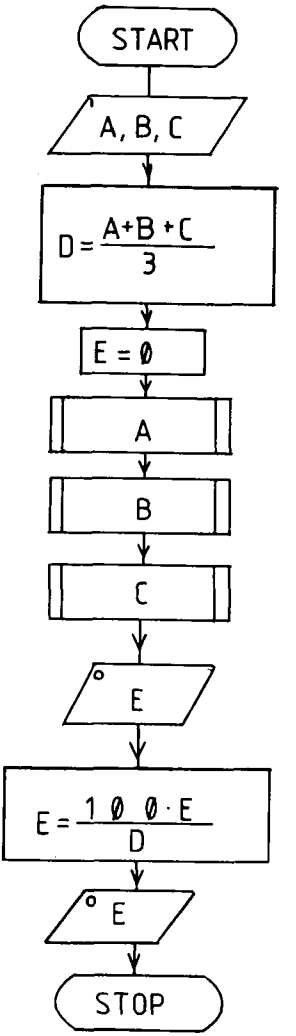
Továbbfejlesztési lehetőségek:

1. A és I kezdőértékét a gép adja.
2. Ugyanaz, mint az előző programnál.

Hibaszámitást végző program

MØ	M1	M2	M3	M4	M5
A	B	C	O	E Ø	1ØØ

```
     LRN
ØØ   RCL 1
     STO 3
     RCL 2
     SUM 3
     RCL Ø
Ø5   SUM 3
     3
     2nd INv Prd 3
     RCL 3
     2nd Pause
1Ø   RCL Ø
     SBR 8
     RCL 1
     SBR 8
     RCL 2
15   SBR 8
     3
     2nd INv Prd 4
     RCL 4
     2nd Pause
2Ø   ÷
     RCL 3
     =
     x
     RCL 5
25   =
     R/S
     2nd LbI 8
     -
     RCL 3
3Ø   =
     2nd |X|
     SUM 4
     INV SBR
     LRN
```



(M. 2.1)

```
Inditás: 1ØØ
         STO 5
         Ø
         STO 4
         A
         STO Ø
         B
         STO 1
         C
         STO 2
         RST
         RIS
```

Ujra inditáskor a 1OO STO 5 elmaradhat, mert az M5 tartalma nem változik a futás alatt.

A program a mérés átlagát, az abszolut hibát és a relativ hibát adja. Fel lehet használni a fizika oktatásában is. A szükös memória kapacitás nem teszi lehetővé, hogy három mérésnél többet vizsgáltassunk.

Osztályátlagot számító program 1.

```
                    MØ  M1  M2
                    I   A   B
    LRN
ØØ  6               1Ø  X
    STO Ø               RCL Ø
    Ø                   =
    STO 1               SUM 2
    STO 2               GTO Ø
Ø5  2nd LbI Ø       15  2nd LbI 1
    2nd INv Dsz         RCL 1
    GTO 1               2nd INv Prd 2
    RIS                 RCL 2
    SUM 1               RIS
                    2Ø  RST
                        LRN
```

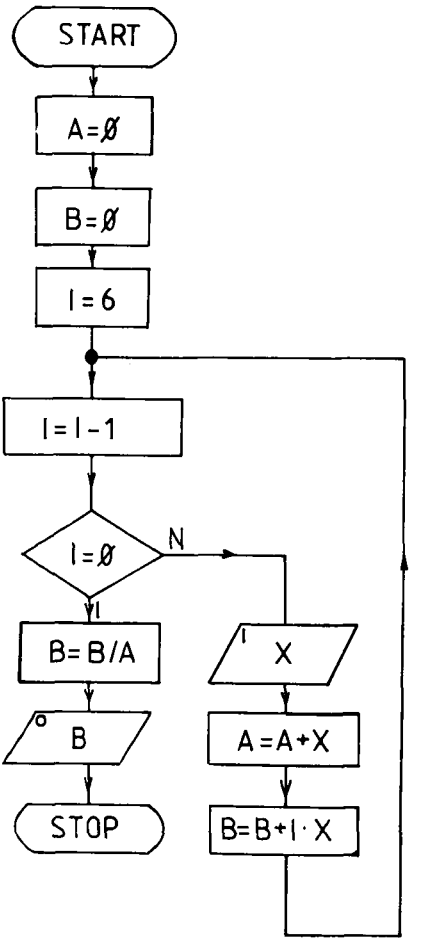
Indítás: RST
R/S

Ujrainditás: R/S

A program indítása után az egyes jegyek számát kell 5-től, csökkenő sorrendben megadni. Példát mutatunk arra, hogy a gép adhatja a változóknak a kezdőértéket. Ennek következtében az indítás nagyon egyszerüvé válik.

A program didaktikai haszna, hogy a gyakorlati életből meritett problémát old meg.

Továbbfejlesztési lehetőség: Vizsgáljuk a kapott adatokat, s amennyiben nem felelnek meg a követelményeknek, adassunk hibajelzést.



(M.2.2)

Osztályátlagot számító program 2.

M0	M1	M2	M3
L	A	B	I
	0	0	0

```
      LRN
00    1          10  2nd LbI 0
      SUM 3          RCL 0
      0              STO 7
      STO 7          RCL 3
      RIS            2nd INV x=t
05    2nd x=t    15  RST
      GTO 0          RCL 1
      SUM 2          2nd INV Drd 2
      1              RCL 2
      SUM 1          RIS
                     LRN
```

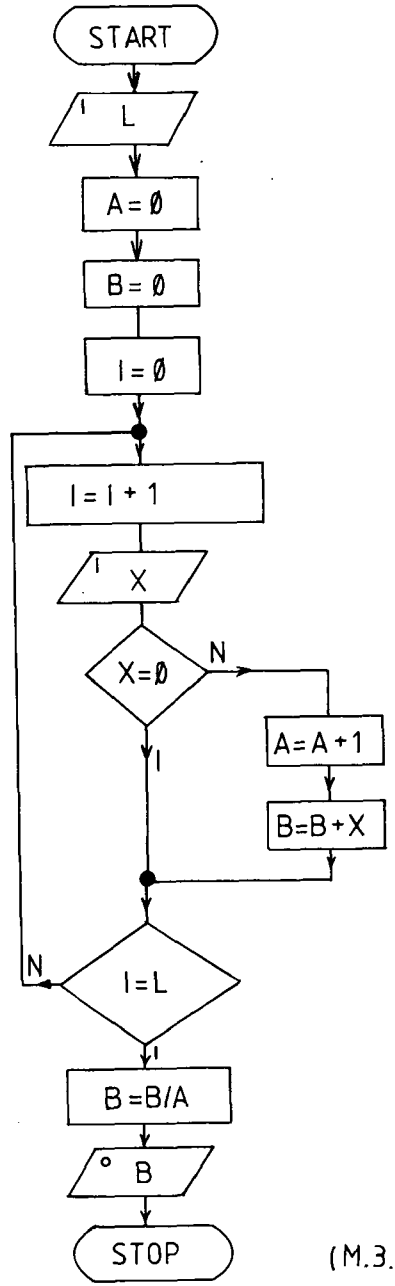
Indítás és ujrainditás:

```
L
STO 0
0
STO 1
STO 2
STO 3
RST
R/S
```

Az osztály létszámának megadása után /L/ az osztályzatokat kell egymás után beadni. Ha a jegy hiányzik, azt a 0 jelzi. A kezdőértékeket nem a gép adja, igy az inditás bonyolultabb, mint az előző példa esetében. A program didaktikai jelentősége hasonló az előző feladatéhoz.

Továbbfejlesztési lehetőségek:

1. A változók állandó kezdőértékeit adja a gép.
2. Ugyanaz, mint az előző feladatnál.



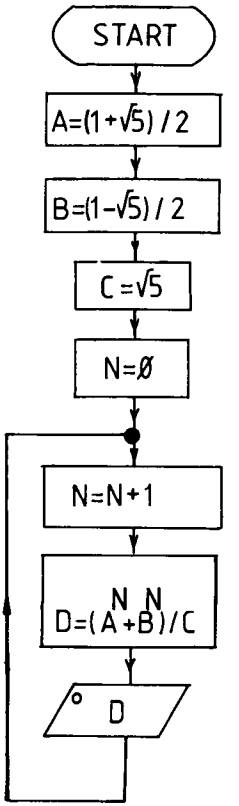
(M.3.1)

MØ	M1	M3	M4
A	B	C	N
$(1+\sqrt{5})/2$	$(1-\sqrt{5})/2$	$\sqrt{5}$	

```
     LRN
ØØ   Ø
     STO 4
     2nd LbI Ø
     1
     SUM 4
Ø5   RCL Ø
     Y^X
     RCL 4
     =
     STO 5
1Ø   RCL 1
     Y^X
     RCL 4
     =
     STO 6
15   RCL 4
     ÷
     2
     =
     2nd INv Int
2Ø   2nd x=t
     GTO 1
     RCL 6
     SUM 5
     GTO 2
25   2nd LbI 1
     RCL 6
     INV SUM 5
     2nd LbI 2
     RCL 3
3Ø   2nd INV Prd 5
     RCL 5
     2nd INV Int
     2nd x=t
     GTO 3
35   RCL 5
     2nd Int
     +
     1
     =
4Ø   STO 5
     2nd LbI 3
     RCL 5
     2nd Pause
     GTO Ø
     LRN
```

Inditás: $\sqrt{5}$, STO 3, $(1+\sqrt{5})/2$, STO Ø, $(1-\sqrt{5})/2$, STO 1, RST, RIS

Ujrainditás: RST, R/S



(M.3.2)

A blokkvázlat a matematikailag bizonyitott képlet alapján készült. Ennek ellenére, ha minden változtatás nélkül programot irunk ennek alapján, helytelen eredményt kapunk. Ennek oka az, hogy

1. a $\sqrt{5}$ értékét a gépünk kerekitve jegyzi meg.
2. negativ alapu hatványozást a gép nem ismer, és

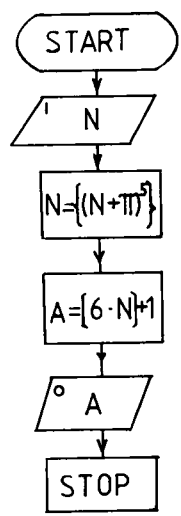
$(1-\sqrt{5})/2 < \emptyset$.

Ezeket a program irásakor figyelembe kell venni. Ez történt a 15-28, valamint a 30-40 lépésekben.

Példát adhatunk arra, hogy helyes algoritmus rossz eredményt adhat. A program jóval bonyolultabb, mint az ugyanezt a feladatot végző korábbiak, csupán az előbb emlitett didaktikai okok miatt közöltük.

A kockadobás modellezése

```
                    MØ
                    N
      LRN
ØØ    RCL Ø         1Ø    6
      +                   =
      2nd π               2nd Int
      =                   +
      Y^X                 1
Ø5    5             15    =
      =                   RIS
      2nd INV Int         RST
      STO Ø               LRN
      X
```



(M.3.3)

Inditás: N
STO Ø
RST
R/S

Ujrainditás: R/S

A véletlenszám generátor, amit alkalmaztunk, a [2] irodalomban található.

A program rendkivül egyszerü, csak azért közüljük, mert egy későbbi programnál felhasználhatjuk.

Miért terjed ki a gáz?

Modellkisérlet [1]

M0	M1	M2	M3	M4	M5
N	A	X	10^{x-1}		10
	654321				

LRN

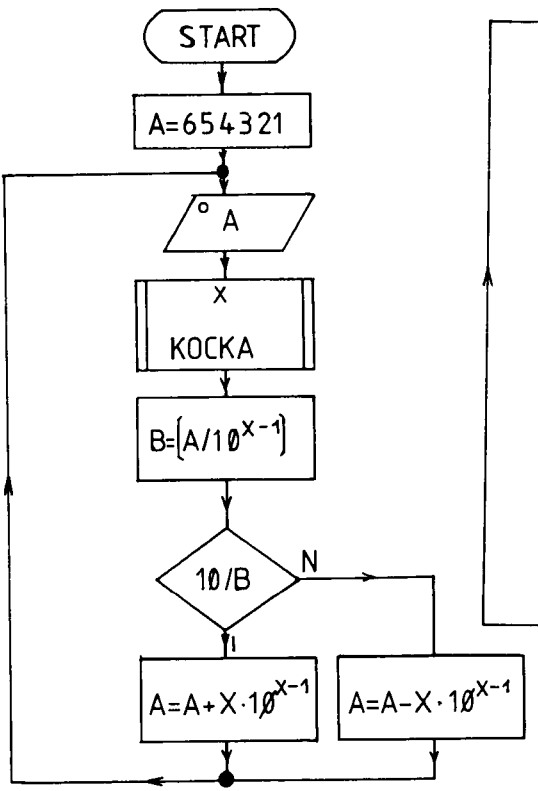
```
00  RCL 1
    RIS
    RCL 0
    +
    2nd π
05  =
    Y^x
    5
    =
    2nd INV Int
10  STO 0
    X
    6
    =
    2nd Int
15  STO 3
    +
    1
    =
    STO 2
20  RCL 5
    Y^x
    RCL 3
    =
    STO 3
25  ÷
    RCL 1
    =
    1/x
    2nd Int
30  ÷
    RCL 5
    =
    STO 7
    2nd Int
35  2nd x=t
    GTO 1
    RCL 2
    X
    RCL 3
40  =
    INV SUM 1
    RST
    2nd LbI 1
    RCL 2
45  X
    RCL 3
    =
    SUM 1
    RST
```

Indítás: N
STO 0
10
STO 5
654321
STO 1
RST
R/S

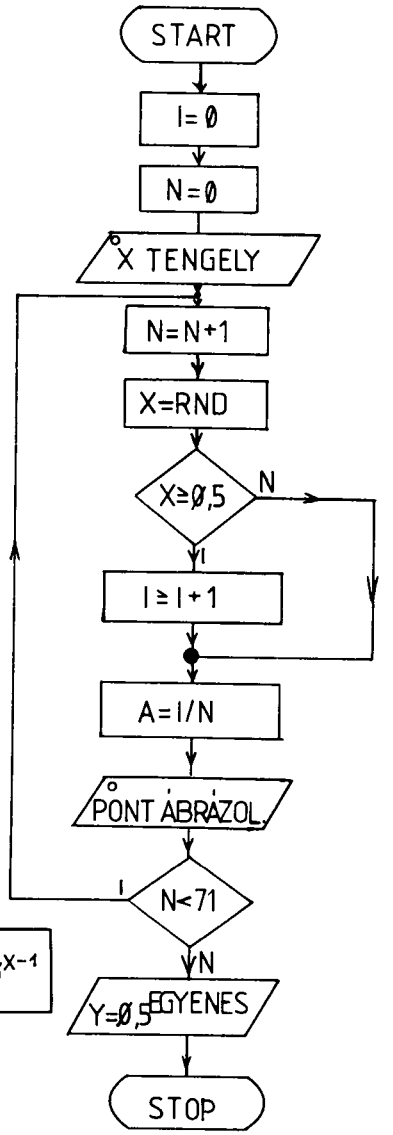
Ujrainditás: 654321
STO 1
RST
R/S

Ez a program elég bonyolult és összetett, ezért csak a kiemelkedő képességü tanulók számára lehet feladat. Minden esetben az A-val jelölt szobában lévő darazsakat jelzi ki. A kockadobást az előző program modellezi.

Az A 0-vá válása esetén a program hibajelzéssel leáll a nullával való osztás miatt. Ez általában elég sok kisérlet után következik be. Ezután a programot ujra inditjuk,


START
A=654321
° A
X
KOCKA
B=[A/10^(X-1)]
10/B
N
I
A=A+X·10^(X-1)
A=A-X·10^(X-1)

(M.4.1)


START
I=0
N=0
° X TENGELY
N=N+1
X=RND
X≥0,5
N
I
I=I+1
A=I/N
° PONT ÁBRÁZOL
I
N<71
N
Y=0,5 EGYENES
STOP

(M. 4. 2)

de ekkor már az A a másik szobát jelenti.

A program didaktikai szempontból nagyon jelentős. Példát adhatunk egy elég bonyolult természeti folyamat számitógépes modellezésére. A tanulók a sok lépés következtében rákényszerülnek arra, hogy próbálják csökkenteni a lépésszámot. A fizikával koncentrációt jelent, valamint a másodikos technika anyag oktatása során hivatkozhatunk rá.

Miért terjed ki a gáz?
Modellkisérlet [1]

```
 M
10 REM DARAZSAK
20 ? CHR$(12)
30 ? CUR(8,3);'MIERT TERJED KI A GAZ?' : ? '
     MODELLKISERLET AZ ELSOS FIZIKAHOZ'
40 OUT 6,5 : FOR I=1 TO 3000 : NEXT I : OUT
   6,0 : ? CHR$(12)
50 ONERRORGOTO 50 : ? 'HANY DOBAST VIZSGALJU
   NK'; : INPUT N
51 IF N<1 THEN 50
60 ONERRORGOTO 60 : ? 'HANY DARAZS VAN(<18)'
   ; : INPUT M
61 IF M<1 OR M>18 THEN 60
70 DIM A%(M),B%(M),C%(M+1)
80 RANDOMIZE
90 FOR I=1 TO M : A%(I)=1 : B%(I)=0 : C%(I)=
   0 : NEXT I
100 C%(M+1)=1
110 ? CHR$(12) : FOR I=1 TO 15 : ? : ? CHR$(
   151); : NEXT I
120 FOR I=5 TO 60 : SETDOT 6,I : SETDOT 42,I
    : NEXT I
130 FOR I=6 TO 42 : SETDOT I,5 : SETDOT I,32
    : SETDOT I,60 : NEXT I
140 ? CUR(17,0);'A KISERLET SORSZAMA:'
150 FOR K=1 TO N : ? CUR(17,24);K
160 FOR I=1 TO M
170 IF A%(I)=1 THEN SETDOT 6+2*I,15 ELSE CLR
   DOT 6+2*I,15
180 IF B%(I)=1 THEN SETDOT 6+2*I,42 ELSE CLR
   DOT 6+2*I,42
190 NEXT I
200 P=INT(M*RND)+1
210 IF A%(P)=1 THEN A%(P)=0 : B%(P)=1 : GOTO
    230
220 B%(P)=0 : A%(P)=1
```

```
230 T=0
240 FOR I=1 TO M
250 IF A%(I)=1 THEN T=T+1
260 NEXT I
270 C%(T+1)=C%(T+1)+1 : OUT 6,2*T+1 : FOR G=
    1 TO 50 : NEXT G : OUT 6,0
280 NEXT K
290 ; CHR$(12)
300 ; '       AZ ELOSZLAS:'
310 ; '       ***********'; : ; : ; ;
320 FOR I=1 TO M+1
330 ; I-1;'   ';C%(I)
340 NEXT I
350 ; ' AZ EREDMENYROL OSZLOPDIAGRAMMOT ADOK
    .'
360 FOR I=1 TO 5000 : NEXT I
370 ; CHR$(12)
380 FOR I=0 TO 15 : ; : ; CHR$(151); : NEXT
    I
390 V%=75/(M+1) : T=0
400 FOR I=1 TO M+1
410 FOR J=45-C%(I) TO 45
420 FOR K=(I-1)*V%+2 TO I*V%
430 IF J<7 THEN T=T+1 : GOTO 440 ELSE SETDOT
    J,K
440 NEXT K : NEXT J : NEXT I
450 IF T<>0 THEN ; CUR(17,0);'AZ ABRA NEM FE
    RT KI A KEPERNYORE.'
460 STOP : END
```

A gondolatmenet lényege megegyezik az előző feladattal. Nagy előny az, hogy több darázs mozgását vizsgálhatjuk. A véletlent az ABC-80-as számitógép véletlenszám generátora modellezi. A tömbök alkalmazása lehetővé teszi mindkét szobában lévő darazsak számontartását.

A grafikus üzemmód alkalmazása láthatóvá teszi a változásokat. Mig az előző programban a statisztika készités a felhasználó feladata volt, itt ezt is a számitógépre bizhatjuk.

A program végén egy oszlopdiagram készitése található. A kisérlet lényegét, a Gauss-görbe létrejöttét mutathatjuk be a segitségével. A nehézségi sorrendben nem itt következne, de kapcsolódik az előző feladathoz, ezért helyeztük el itt.

A pénzfeldobás modellezése

```
5 RANDOMIZE
10 REM PENZFELDOBAS
20 FOR I=0 TO 23 : ? : ? CHR$(151); : NEXT I

25 FOR I=1 TO 70 : SETDOT I,2 : NEXT I
30 FOR I=3 TO 70 : SETDOT 70,I : NEXT I
80 I=0 : N=0
90 N=N+1
100 X=RND
110 IF X).5 THEN I=I+1
130 A=70*I/N : A=INT(A)
150 SETDOT 70-A,N+2 : OUT 6,2*A+1 : FOR T=1
    TO 80 : NEXT T : OUT 6,0
170 IF N(71 THEN 90
180 FOR J=2 TO 70 STEP 2 : OUT 6,3 : SETDOT
    35,J : FOR T=1 TO 30 : NEXT T : NEXT J
    : OUT 6,0
```

A valószinüségszámitás alapkisérletét modellezi a program. Gyakoroltatható vele a grafikus üzemmód alkalmazása. Előnye, hogy egyből ábrázolja a relativ gyakoriságot a kisérletek függvényében. Ha e program nélkül, a valóságban végeztetjük a kisérletet, nagyon sok időt vesz igénybe. A kisérletszám korlátozott a képernyő felbontása miatt.
A program továbbfejleszthető ugy, hogy amikor a képernyő megtelik, letörli, és ugy folytatja tovább a program futását.

Naptár készitő program

```
10 REM NAPTAR KESZITES
20 ? CHR$(12) : ? CUR(10,5);'     NAPTAR KES
   ZITESE'
30 OUT 6,5 : FOR I=1 TO 2000 : NEXT I : OUT
   6,0
40 DIM A$(12),N(12),B$(7),M(7,6),Q(7)
```

```
50 FOR I=1 TO 12 : READ A$(I) : NEXT I
60 DATA JANUAR,FEBRUAR,MARCIUS,APRILIS,MAJUS
   ,JUNIUS,JULIUS,AUGUSZTUS,SZEPTEMBER,OKT
   OBER,NOVEMBER,DECEMBER
70 FOR I=1 TO 7 STEP 2
80 N(I)=31 : NEXT I
90 FOR I=8 TO 12 STEP 2 : N(I)=31 : NEXT I
100 N(4)=30 : N(6)=30 : N(9)=30 : N(11)=30
110 FOR I=1 TO 7 : READ B$(I) : NEXT I
120 DATA HETFO,KEDD,SZERDA,CSUTORTOK,PENTEK,
   SZOMBAT,VASARNAP
130 FOR I=1 TO 7 : FOR J=1 TO 5 : M(I,J)=0 :
    NEXT J : Q(I)=0 : NEXT I
140 ONERRORGOTO 140 : ; 'MELYIK EVBOL KER NA
   PTART(1901-2099)' : INPUT E
150 IF E)2099 OR E(1901 OR INT(E)()E THEN 14
   0
160 ; 'MELYIK HONAP NAPTARAT KERI'; : INPUT
   C$
170 I=0
180 I=I+1
190 IF C$()A$(I) THEN IF I(12 THEN 180 ELSE
   160
200 IF I=2 THEN 410
210 IF I=1 THEN G=E-1 : J=14
220 IF I=2 THEN G=E-1 : J=15
230 G=E : J=I+1
240 FOR T=1 TO N(I)
250 Y$=MUL$('365.25',NUM$(G),2) : L=LEN(Y$)
    : S=L-3 : Y$=LEFT$(Y$,S)
260 X$=MUL$('30.6',NUM$(J),1) : L=LEN(X$) :
   X$=LEFT$(X$,L-2)
270 Y$=ADD$(Y$,X$,0) : Y$=SUB$(Y$,'694066',0
   ) : Y$=ADD$(Y$,NUM$(T),0)
280 Y$=DIV$(Y$,'7',4) : L=LEN(Y$) : X$=LEFT$
   (Y$,L-5) : Y$=SUB$(Y$,X$,4) : Y$=MUL$(Y
   $,'7',4) : Y=VAL(Y$)
290 IF Y-INT(Y))INT(Y)+1-Y THEN Y=INT(Y)+1 E
   LSE Y=INT(Y)
300 IF Y=0 THEN Y=7
310 Q(Y)=Q(Y)+1
320 M(Y,Q(Y))=T
330 IF T=1 AND Y()1 THEN FOR R=1 TO Y-1 : Q(
   R)=Q(R)+1 : NEXT R
340 NEXT T
350 ; CHR$(12) : ; ' NAPTAR AZ ';E;'. EV ';C
   $;' HONAPJARA'
360 ; '*************************************
   *'
```

```
370 FOR I=1 TO 7
380 ; B$(I);TAB(15);
390 FOR J=1 TO 6 : ; M(I,J); : NEXT J
400 ; : NEXT I : STOP
410 IF INT(E/4)=E/4 THEN 430
420 N(2)=28 : GOTO 210
430 IF INT(E/100)<>E/100 THEN N(2)=29 : GOTO
    210
440 IF INT(E/400)=E/400 THEN N(2)=29 : GOTO
    210
450 GOTO 420
```

Az összetettsége, bonyolultsága miatt ezt a feladatot csak jó képességü tanulók számára ajánljuk.
A müködése a [3] irodalomban található öröknaptár képleten alapszik.
Az ABC-80-as számitógépen levő egyszerü aritmetikai müveletek alkalmazásakor a program nem müködött megfelelően. Ennek oka az, hogy a kerekitések elrontották az eredményt. Nagy előnyét mutathatjuk meg a gépnek e program segitségével. Ha az ASC II. aritmetikát alkalmazzuk, a program müködőképessé válik.
A tetszetős táblázatkészitést is tanulmányozhatják a tanulók ezen program segitségével. Ezt a tudást egyéb programok esetén is hasznosithatják.

cimke	a memóriarekesz tartalma
Ø Ø Ø	Ø Ø 5 Ø 1 5 Ø 3
Ø Ø 1	Ø 2 5 Ø Ø 5 Ø 1
Ø Ø 2	Ø 5 5 Ø 1 5 Ø 2
Ø Ø 3	Ø 6 - - - - - -
Ø Ø 4	Ø 4 Ø Ø Ø - - -
5 Ø Ø	Ø Ø Ø Ø Ø Ø Ø 1
5 Ø 1	Ø Ø Ø Ø Ø Ø Ø Ø
5 Ø 2	Ø Ø Ø Ø Ø Ø Ø 5
5 Ø 3	Ø Ø Ø Ø Ø Ø Ø 1

Megjegyzések: 1. A - helyére bármilyen szám kerülhet
2. Az adatregiszterekben csak a kezdeti a-

datok szerepelnek.

IRODALOM

[1] BALÁNYI M. - FODOR E. - MARX Gy. - SARKADI I. - TÓTH E. UJJ J.: Fizika I. Tankönyvkiadó, Budapest, 1981

[2] RAY, C.: Kulcs a mikroszámitógépekhez, Müszaki Könyvkiadó, Budapest, 1983

[3] CSÁKÁNY A.: Mit tud a zsebszámológép? Müszaki Könyvkiadó, Budapest, 1981

[4] A. G. KURCS: Felsőbb algebra. Tankönyvkiadó, Budapest, 1975

[5] SALÁNKI J.: A számitástechnika alapjai. Tankönyvkiadó, Budapest, 1981

[6] SZÜCS E.: Technika a gimnázium I. osztálya számára Tankönyvkiadó, Budapest, 1981

[7] SZÜCS E.: Technika a gimnázium II. osztálya számára Tankönyvkiadó, Budapest, 1982

[8] Természettudományi lexikon II. kötet. Akadémiai Könyvkiadó, Budapest

[9] ZÖLD S.: Harmadik generációs számitógép alapismeretek KSH Nemzetközi Számitástechnika Oktató- és Tájékoztató Központ, Budapest, 1976

THE TEACHING OF COMPUTING TECHNIQUES IN TECHNOLOGY LESSONS

by

Tamás Tarcsay

Summary

Attention is drawn to a possibility for the teaching of computing techniques in grammar schools. On the basis of his teaching experience, the author shows that the present technology syllabus provides a possibility for teaching the fundamentals of computing techniques.

The technology teaching matter in the first two years of the grammar school is examined; attention is paid to the main points of interconnection of computing techniques and general technology; and the computational material that can be taught to children of this age is discussed.

Primarily the programming /software/ is dealt with, but mention is also made of the importance of the hardware knowledge necessary for this.

The Appendix contains a collection of programs that can be used in the lessons; it gives programs that can be prepared. These programs follow one another in sequence of increasing difficulty, this facilitating their use in the lessons. A block diagram is given for each program. Use of these better illustrates the discussed algorithm.